APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/997,990	11/30/2001	Jeremy Alan Arnold	IBM / 193	4258

7590 08/03/2004	EXAMINER
Scott A. Stinebruner	PHAM, CHRYSTINE
Wood, Herron & Evans, L.L.P.	

ART UNIT	PAPER NUMBER
2122	

Scott A. Stinebruner
Wood, Herron & Evans, L.L.P.
2700 Carew Tower
441 Vine St.
Cincinnati, OH 45202-2917

DATE MAILED: 08/03/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) FROM
THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
  after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
  earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>30 November 2001</u>.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>1-40</u> is/are pending in the application.

     4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) <u>1-40</u> is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on <u>30 November 2001</u> is/are: a)☒ accepted or b)☐ objected to by the Examiner.

     Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

     Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

     a)☐ All   b)☐ Some * c)☐ None of:

         1.☐ Certified copies of the priority documents have been received.

         2.☐ Certified copies of the priority documents have been received in Application No. _____.

         3.☐ Copies of the certified copies of the priority documents have been received in this National Stage
application from the International Bureau (PCT Rule 17.2(a)).

     * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
     Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
     Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Specification

1. The attempt to incorporate subject matter into this application by reference to related patent application (page 15 line 19-23) is improper because it fails to provide updated status (i.e., serial/patent numbers) for the related patent application. Appropriate correction is required.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –
   (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1-9, 12-13, 23-26, 28-29, 33, and 38-39 are rejected under 35 U.S.C. 102(b) as being anticipated by West (US 5740440), hereinafter *West*.

As per claim 1, *West* teaches a computer-implemented method (program product) (e.g., col.6:34-36, col.37:34-40) of debugging an object-oriented computer program (e.g., see Abstract & col.1:5-8, see *monitor program 44* FIG.2 & associated text), the method comprising:

- o in response to user input, identifying a plurality of creators for a class defined in the object-oriented computer program (e.g., col.11:37-41, col.23:17-18, col.34:23-24) and setting a plurality of breakpoints on the identified creators (e.g., col.4:1-4, col.35:9-11) ; and

- o halting execution of the object-oriented computer program during debugging in response to hitting any of the plurality of breakpoints (e.g., col.3:63-64, col.4:4-5, col.35:12-15, FIG.5 *60* & *72*).

As per claim 2, *West* teaches the method as applied to claim 1, wherein identifying the plurality of creators includes identifying every creator for the class (e.g., col.11:36-47, col.12:33-37, col.23:16-18).

As per claim 3, *West* teaches the method as applied to claim 1, further comprising, after identifying the plurality of creators, displaying a list of the identified creators (e.g., col.4:9-10, col.8: 24-30, FIG.2 *user interface subsystem 42* & associated text, col.23:17-18, see *BinaryNode::BinaryNode* FIG.16 & associated text, col.21:59-63, FIG.18 & associated text) and receiving user input to select a subset of identified creators (e.g., col.22:1-2 & 23-25, col.34:18-23), wherein the plurality of breakpoints are set on only the subset of the identified creators (e.g., col.4:1-4) .

As per claim 4, *West* teaches the method as applied to claim 1, wherein the plurality of breakpoints are collectively set on all of the identified creators in response to the user input (e.g., see *set_breakpoints();* col.26:54-65).

As per claim 5, *West* teaches the method as applied to claim 1, wherein setting the plurality of breakpoints includes setting each breakpoint from plurality of breakpoints on a statement in one of the identified creators (e.g., col.9:44-46).

As per claim 6, *West* teaches the method as applied to claim 5, wherein setting each breakpoint includes inserting debugging program code in the creator on which such breakpoint is set (e.g., col.10:36-49, col.33:9-12).

As per claim 7, *West* teaches the method of claim 1, wherein identifying the plurality of creators and setting the plurality of breakpoints are performed in response to user input to set a

creation breakpoint, and wherein the plurality of breakpoints are associated with the creation

breakpoint (e.g., col.10:50-55 & 59-62).

As per claim 8, *West* teaches the method as applied to claim 7, further comprising, in

response to the user input to set the creation breakpoint, adding an entry for the creation

breakpoint in a breakpoint data structure, wherein setting the plurality of breakpoints includes

storing breakpoint information for each breakpoint in the breakpoint data structure, wherein the

breakpoint information for each breakpoint is associated with the entry in the breakpoint data

structure for the creation breakpoint (e.g., see *BreakPoint instance 264* col.10:62-67, col.3:53-57,

FIG.5 & associated text, col.12:66-col.13:6).

As per claim 9, *West* teaches a method as applied to claim 1, wherein a total number of

hits to the plurality of breakpoints are tracked and processing is done on said breakpoints to

generate updates (e.g., see *Break Point 264 & Low Level Event 259 & Object Event Set 263 &*

Object Event 261 & Object Event List 260 & Method Call 259 & Method Call Stack 256 FIG.4 &

associated text, col.12:50-55, col.13:37-49, see step *60-68* FIG.5 & associated text, see step *78-*

84 FIG.6 & associated text).

As per claim 12, *West* teaches the method as applied to claim 1, wherein each creator

comprises a constructor method defined in the class (e.g., col.7:5-10).

As per claim 13, *West* teaches the method as applied to claim 1, further comprising

collectively removing the plurality of breakpoints in response to user input (e.g., see

remove_breakpoints(); col.26:54-65, col.4:47-48).

As per claim 23, *West* teaches an apparatus, comprising:

- o   a memory within which resides at least a portion of an object-oriented computer program (e.g., col.12:30-32, see *RAM 14 & ROM 16* FIG.1 & associated text); and

- o   program code configured to debug the object-oriented computer program by, in response to user input, identifying a plurality of creators for a class defined in the object-oriented computer program and setting a plurality of breakpoints on the identified creators, and halting execution of the object-oriented computer program during debugging in response to hitting any of the plurality of breakpoints (see claim 1).

As per claims 24-26, 28-29, and 33, they recite limitations which have been addressed in claims 2-4, 7-8, and 13 respectively, therefore, are rejected for the same reasons as cited in claims 2-4, 7-8, and 13.

As per claim 38, *West* teaches a program product (e.g., col.6:34-36, col.37:34-40), comprising:

- o   program code configured to debug an object -oriented computer program by, in response to user input, identifying a plurality of creators for a class defined in the object-oriented computer program and setting a plurality of breakpoints on the identified creators, and halting execution of the object-oriented computer program during debugging in response to hitting any of the plurality of breakpoints (see claim 1); and

- o   a signal bearing medium bearing the program code (e.g., see *RAM 14 & ROM 16* FIG.1 & associated text).

As per claim 39, *West* teaches the program product of claim 38, wherein the signal bearing medium includes at least one of a transmission medium (e.g., see *bus 12 & COMM 34* FIG.1 & associated text) and a recordable medium (e.g., see *RAM 14* FIG.1 & associated text).

Claim Rejections - 35 USC § 103

4.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness

rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this
> title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would
> have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

5.   Claims 10-11, 14-22, 27, 30-32, 34-37, and 40 are rejected under 35 U.S.C. 103(a) as being

unpatentable over *West* in view of applicant's Admission of Prior Art (hereinafter APA – see

Background of the Invention pg.2 line 3-8, pg.3 line 1-8 and 20-22).

As per claim 10, *West* teaches the method as applied to claim 9, wherein halting

execution of the object-oriented computer program during debugging responses to hitting any of

the plurality of breakpoints (see claim 1). *West* does not expressly disclose determining whether

the total number of hits meets a condition in response to hitting any of the plurality of breakpoints

and halting execution of the object-oriented computer program if the total number of hits meets

the condition. However, APA discloses a method of debugging an object-oriented computer

program comprising tracking a total number of hits to the plurality of breakpoints (e.g., pg.2 line 7-

8), determining whether the total number of hits meets a condition in response to hitting any of

the plurality of breakpoints and halting execution of the object-oriented computer program if the

total number of hits meets the condition/threshold (e.g., see *X times* pg.2 line 5-8). It would have

been obvious to one of ordinary skill in the pertinent art at the time the invention was made to

incorporate the teaching disclosed by APA with the teaching of *West* and the motivation for doing

so would have been that determining and halting execution only when the total number of hits for

a breakpoint A meets a condition/threshold (e.g., X times) would allow the user of the debugging

method more control over program suspension. That is to say, by defining and monitoring said

condition/threshold associated with breakpoint A, the user could skip over X-1 breakpoint

iteration(s) (which entail X-1 suspensions of program execution) and jump to the breakpoint of interest (i.e., when breakpoint A has been hit X times), thus making program debugging a more effective and efficient process.

As per claim 14, *West* teaches a computer-implemented method (program product) of debugging an object-oriented computer program (see claim 1), the method comprising tracking the number of object creations of a class defined in the object-oriented computer program (e.g., see *item 1, item 2, item 3* FIG.21 & associated text, see *R0, R1, R2* FIG.22 & associated text, col.4:15-22, col.14:1-2 & 17-19, col.18:30-38,col.19:49-55, FIG.20 & associated text, col.29:12-25, col.31:59-61). *West* does not expressly disclose halting execution of the object-oriented computer program in response to the number of object creations meeting a condition. However, APA discloses a method of tracking the number of hits for breakpoints **which are set in class constructors** (emphasis added) and halting execution of the program in response to the number of hits meeting a condition (see claim 10). It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made that tracking the number of hits for breakpoints (see claim 9) which are set in class constructors was analogous to tracking the number of object creations since a class constructor is invoked each time an object is created, causing the breakpoint(s) embedded therein to be hit and tracked. Therefore, the motivation for combining the two teachings of *West* and APA would have been the same as has been cited in claim 10.

As per claims 11, 15, and 16, they recite limitations which have been addressed in claim 10, therefore, are rejected for the same reasons as cited in claim 10.

As per claim 17-22, they recite limitations which have been addressed in claims 1-4, 7, and 12, therefore, are rejected for the same reasons as cited in claims 1-4, 7, and 12.

As per claim 27, 30-32, 34-37, and 40, they recite limitations which have been addressed in claims 1, 5, 7, 9-10, 14, 23, and 38, therefore, are rejected for the same reasons as cited in claims 1, 5, 7, 9-10, 14, 23, and 38.

Conclusion

6.   The prior art made of record and not relied upon is considered pertinent to applicant's disclosure:

- Debugger using class information and dynamic instance inter-relationships, Nishimura et al. (US 5845125)

- Apparatus, program product and method of debugging utilizing a context sensitive breakpoint, Bates et al. (US 6077312)

- System for graphically representing operation of object-oriented programs, Beck et al. (US 4885717)

- Generating symbolic debug information by merging translation and compiler debug information, Lenkov et al. (US 5560009)

- Program debugging system for debugging a program having graphical user interface, Inoue (US 5926638)

- Tracker class for object-oriented programming environments, Wang (US 5940616)

- Debug interface including logic generating handshake signals between a processor, an input/output port, and a trace logic, Madduri (US 6189140)

- Method and system for analyzing and displaying program information, Hamada et al. (US 6240549)

- Visualization in a modular software system, Nwana et al. (US 6266805)

- Object oriented monitor displaying interactions among objects in square matrix with an object designated in first column and additional column(s) for interacting objects, Hurd II et al. (US 6266806)

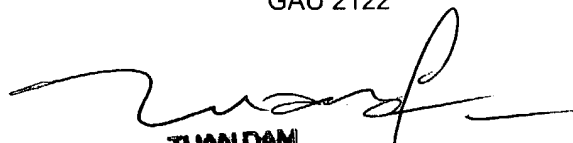- Goal-directed object-oriented debugging system, Wimble et al. (US 5778230).

7.   Any inquiry concerning this communication or earlier communications from the examiner should

be directed to Chrystine Pham whose telephone number is 703.605.1219. The examiner can

normally be reached on Mon-Fri from 8:30am to 5pm.

        If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Tuan Q Dam, can be reached on 703.305.4552. The fax phone number for the

organization where this application or proceeding is assigned is 703-872-9306.

        Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR system,

see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Chrystine Pham
Examiner
GAU 2122

TUAN DAM
SUPERVISORY PATENT EXAMINER